Building automation project using existing IHC-output modules

Overview

The goal of this project is to design and implement a system that uses existing IHC smart home output modules to control lights and some other targets.

Reporting is done according to Centria thesis guidance.

IHC building automation system – a simplified system structure

Figure 1 describes a simplified IHC-configuration that is enough to understand part of the requirements for the project.

The main objective is to make it possible to control the output modules without the IHC controller. To design and implement such a solution we need to understand the control protocol between ICH-controller and output modules. Also the physical and electric connections and safety issues must be well understood.

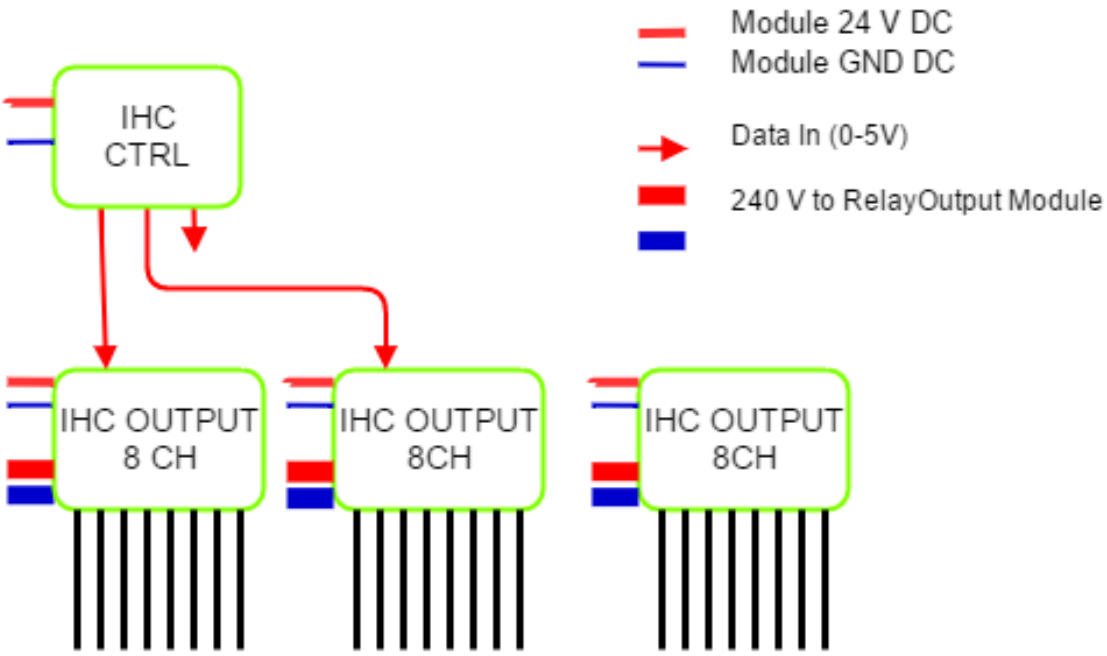

Figure 1 Simplified IHC system structure - input modules omitted

IHC module control – bit encoding

Figure 2 describes the protocol that is used to control each ICH output module. Output module recognizes the start of control sequence from a 4100 us start pulse followed by 300 us low pulse. After that there is 600 us control pulse for each channel which start always with high state. If the duration of the high state is 300 us then the pulse is interpreted as zero and the channel relay will be put off. If the duration of the high state is 150 us then the pulse is interpreted as one and the corresponding relay will be put on.

The parity bit should have value zero if the sum of preceding bit is even. Otherwise its value should be one.

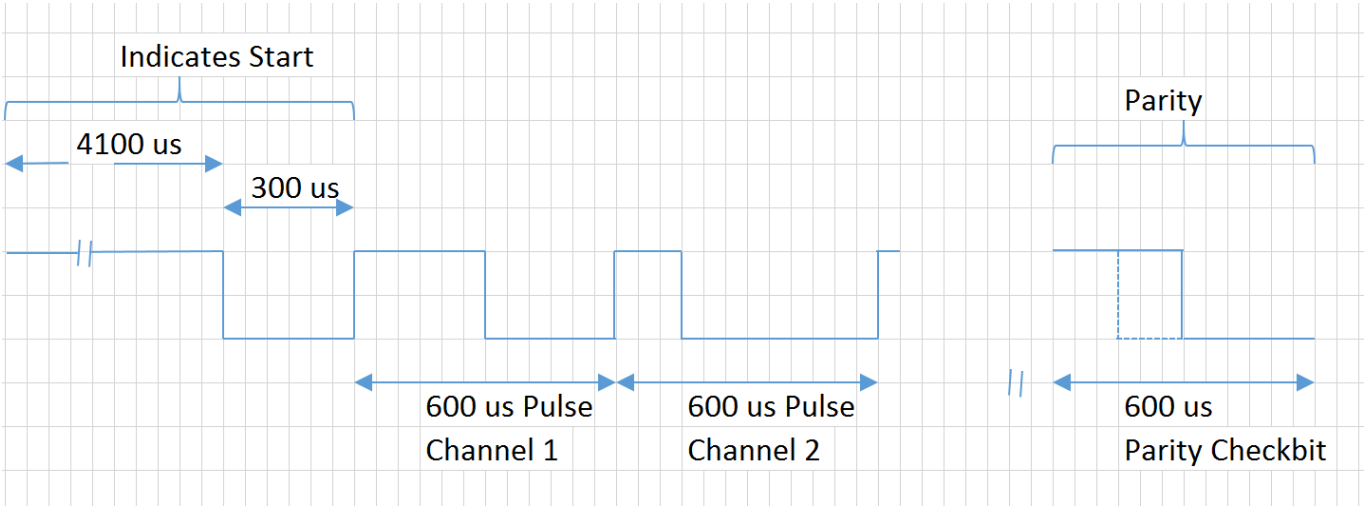

Figure 2 ICH Module Output Module Control Protocol – voltage levels are between 0 – 5 V

The protocol is proprietary and the information presented above is based on unofficial sources (/1/, /2/). Source /2/ gives some hints of the timing tolerances.

Possible system structures for an alternative IHC module control

Figure 3 presents one possible idea for an alternative way to control IHC-output modules. The main idea in this solution is that the user connects with browser to a web server that loads an HTML5-based user interface. The web server provides a REST-API for the browser. The REST API have the functions needed to command the IHC-modules via a real time IHC module driver, RTU MCU in the picture. Between the web server and the RT MCU there must be some bridging software.

The dotted rectangle in Figure 3 suggests that the HW-solution can be one physical device, e.g. Arduino Yun or two distinct systems, e.g. microcontroller board + raspberry Pi.

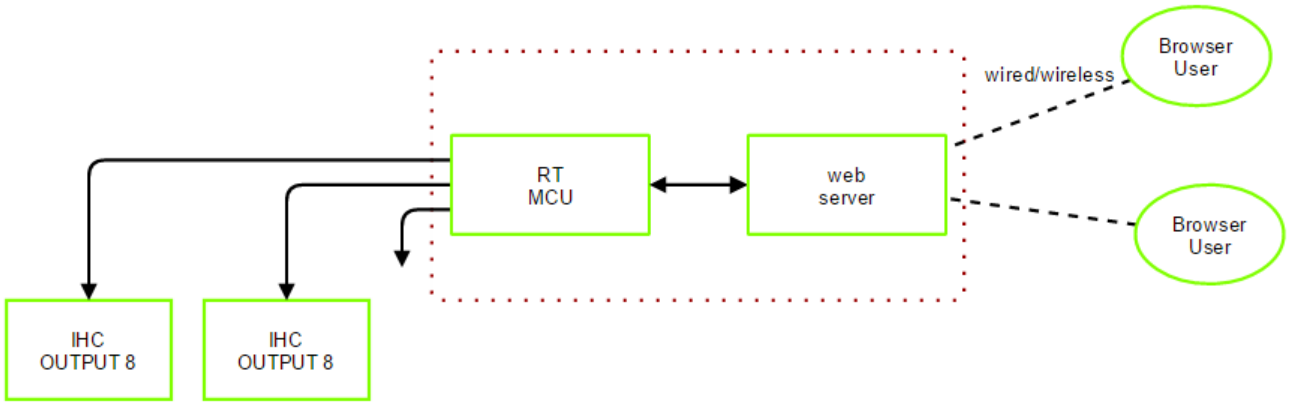

Figure 3 Possible structure for the alternate IHC module controller.

An alternative way to describe the overall structure of the solution described above is presented in Figure 4 concept map. It shows the main logical and physical parts of the solution and their relations.

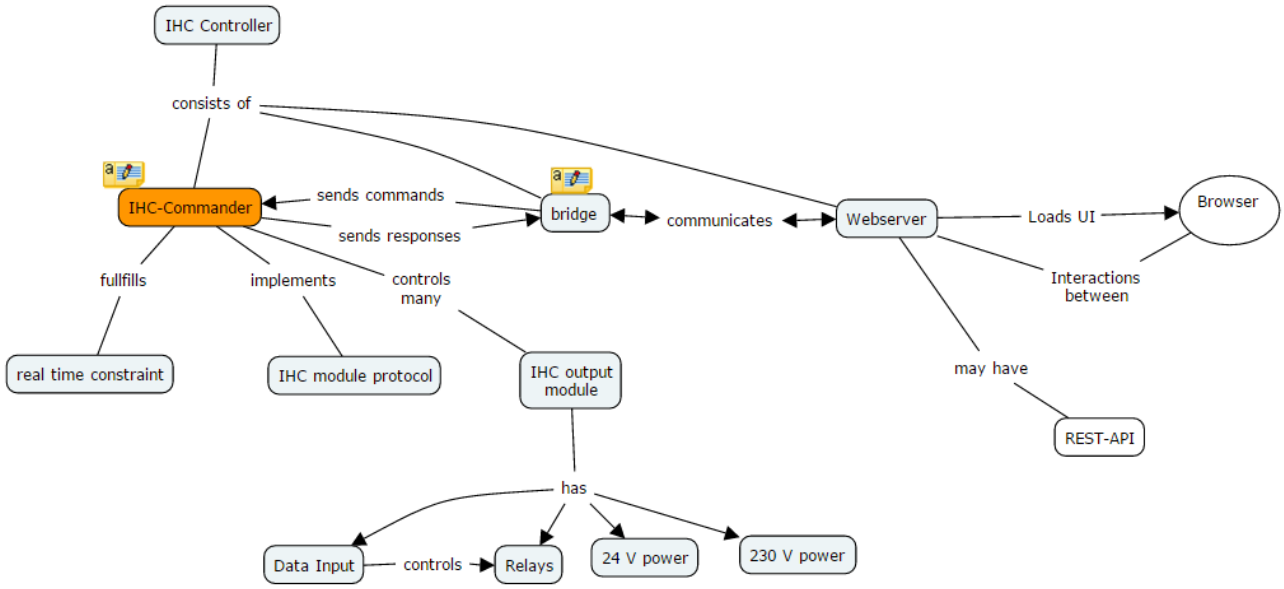

Figure 4 Concept map that may clarify the structure

Requirements for the solution

The end user should be able to see the current state of controllable electric devices like lamps and power sockets. Some of the lamps are dimmable and the user should be able to control the brightness at least as easily as with a physical button. The UI could be location aware and its configuration could be e.g. xml based. *For the initial test purposes the simplest UI could consists of a spinner like control to select the room or area and another one to select the target in room. Then user selects one of the available controls operation against the selected target that is most often ON or OFF.*

The solution should be designed and constructed to fulfil the qualities presented in Figure 5 at an agreed level. Some other qualities may be added during the project.

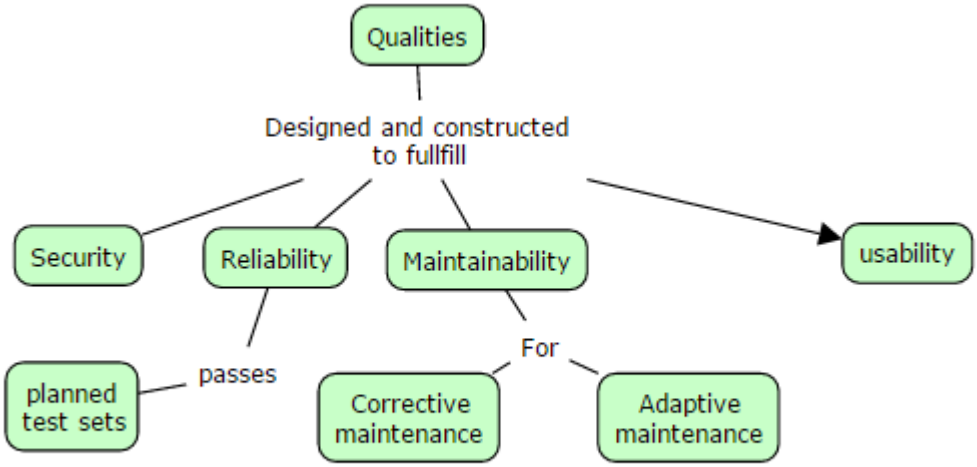

Figure 5 Most essential qualities of the solution

Challenges and possibilities

You have good enough combined skill set to complete successfully this project. The challenge is to organise the work in such a way that subgroups and individuals can work in an efficient and effective way without unplanned delays.

For many subparts of the system there are alternative solutions you can compare and contrast. The report is the correct place to argument your solutions.

The amount of code and dependencies should be kept minimal without sacrificing functionality and usability.

Short review of current trends and standards in the area of home automation and related topics could be taken into theory part of report, e.g. two pages.

Miscellaneous

Dimmer

Dimmer works in the following way: if the control input (3) is on for less than 0.5 seconds the power comes on at the same level it was in in the last time when it was shut. If the duration is more than 0.5 seconds the power level ramps up and down – when released the power levels remains on the values at the moment of release. One more short (< 0.5 s) shuts power off.

The description above assumed that the power output is controlled using 230 V control input (3). The other option is to use low voltage control inputs (1, 2).

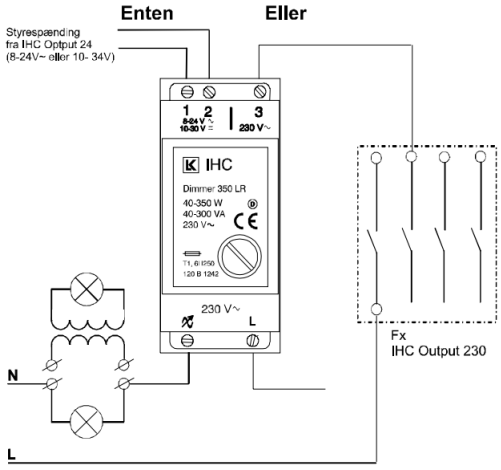

Figure 6 Dimmer - excerpt from Danish language product description

The text beginning here may be moved later to more appropriate context in this document. To operate the dimmer we need an ON-command that has a duration parameter.

Real-time part

If an Arduino board is selected as an ICH commander‘s implementation platform then students need to study how to write c-code to the board bypassing totally or partially the Arduino default software structure. E.g. how to take benefit of the Atmel Studio and how to load code to MCU.

Reprogramming should be possible also later when the system is taken into use – in other words the IHC Commander should be field programmable. Part of the firmware could even be update through the command interface (state machines reconfiguring).

The software must be implemented using Time Triggered Architecture approach where only timer interrupt is allowed. ICH output module pulsing and communication with the non-real-time part are done in their own “tasks” that are called periodically in the timer ISR, see Figure 7

Watchdog functionality is also needed and bookkeeping of non-desired resets can be queried from the module.

Initializations must be well specified – what outputs should be off and what on after power ups and resets.

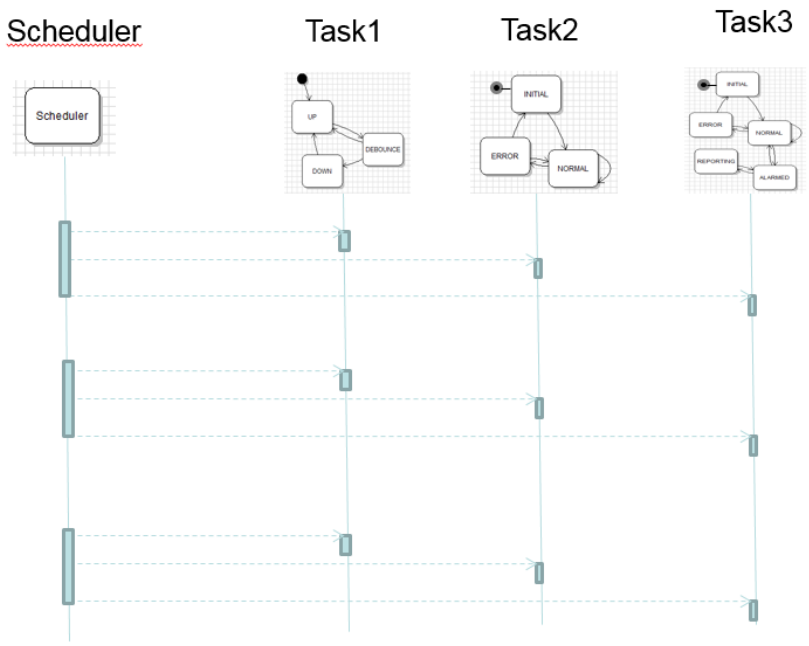

Figure 7 Time Triggered Architecture example

TBD – could the one interrupt source rule loosened if the serial interrupts are disabled during output module pulsing – this makes 5 ms latency to serial processing. This may lead to corrupted messages in receive that are observed from the CRC. This would require application level retransmission in the non-real-time part when confirmation is note got in time or when negative confirmation is got. The latter option would bet better.

Exercise: Design and implement a polling based serial task that stores the received data in a buffer that is sent back in another task. Initializations are done in the common initialization section.

Note: the normal cycle of events in IHC Commander is: Wait for command, Execute Command (Only Scheduler Clock interrupts enabled), Send Response, reset the WD, New Cycle begins.